

Name: Anushka Harshavadan Nevgi

Experiment: 3. Implementation of lexical analysis, input as c file and create keywords, operator, special Symbol, identify file from given file.

Class: TY CSE A, 31

CODE(INPUTFILE):

```
#include
<iostream>
using namespace std
; int main() {
    int a, b, sum;
    cout << "Enter two integers:";
    cin >> a >> b;
    sum = a +
    b;
    cout << sum
; return 0;}
```

FILEHANDLING:

```
#include<iostream>
#include <fstream>
#include <string>
#include <sstream>
#include <set>
#include <cctype>

using namespace std;

bool isKeyword(const string& word) { const
    set<string> keywords = {
        "int", "return", "cout", "using", "namespace",
        "main", "include"};
    return keywords.find(word) != keywords.end();
}

bool isSpecialCharacter(char c) {
    return ispunct(c) && c != '_' && c != '$';
}

bool isIdentifier(const string& word) { if
    (isdigit(word[0])) return false;
    for (char c: word) {
        if (!isalnum(c) && c != '_' && c != '$') return false;
    }
    return true;
}

int main() {
    ifstream infile("code.cpp");
    if (!infile) {
        cerr << "Error opening code.cpp file!" <<
endl;
```

```
return 1; }
```

```
int keywordCount = 0, identifierCount = 0,
specialCharCount = 0, operatorCount = 0;
```

```
ofstream keywordFile("keywords.txt");
ofstream identifierFile("identifiers.txt");
ofstream
```

```
specialCharFile("special_characters.txt");
ofstream operatorFile("operators.txt");
```

```
if (!keywordFile || !identifierFile ||
!specialCharFile || !operatorFile) {
    cerr << "Error opening output files!" << endl;
    return 1; }
```

```
string word;
```

```
char ch;
```

```
while (infile.get(ch)) {
    if (isspace(ch) || ispunct(ch)) { if
        (!word.empty()) {
            if (isKeyword(word)) {
                keywordFile << word << endl;
                keywordCount++;
            } else if (isIdentifier(word)) {
                identifierFile << word << endl;
                identifierCount++;
            }
            word.clear();
        }
    }
```

```
if (isSpecialCharacter(ch)) {
    specialCharFile << ch << endl;
```

```

        specialCharCount++;    }

        if(ch=='+'|| ch=='-'|| ch=='*'|| ch=='/'
|| ch == '=' || ch == '<' || ch == '>') {
            operatorFile<<ch<<endl;
            operatorCount++;    }
        }else{
            word +=ch;    }}

    if(!word.empty()){
        if (isKeyword(word)) {
            keywordFile<<word<<endl;
            keywordCount++;
        } else if (isIdentifier(word)) {
            identifierFile<<word<<endl;
            identifierCount++;    }}

    infile.close();
    keywordFile.close();
    identifierFile.close();
    specialCharFile.close();
    operatorFile.close();

    cout<<"Countswrittentofilessuccessfully."
<<endl;
    return 0;
}

```

OUTPUT:

```

Counts written to files successfully.

-----
Process exited after 0.3539 seconds with return value 0
Press any key to continue . . .

```

operators - Notepad

File Edit Format View Help

```

k
>
/
/
<
<
>
>
>
>
/
/
=
+
<
<

```

keywords - Notepad

File Edit Format View Help

```

include
using
namespace
int
main
int
cout
cout
return

```

identifiers - Notepad

File Edit Format View Help

```

iostream
std
a
b
sum
Enter
two
integers
cin
a
b
sum
a
b
sum

```

special_characters - Notepad

File Edit Format View Help

```

# < > ; ( ) { / / , , ; < < " : " ;
> > > > ; / / = + ; < < ; ; }

```